



# Java Futures Preview

We're not done yet!

**Chris Filachek**

z/TPF and z/TPFDF Architecture & Development

© 2017 IBM z/TPF | TPF Users Group Spring Conference | IBM Confidential

IBM **z/TPF**  
April 4, 2017

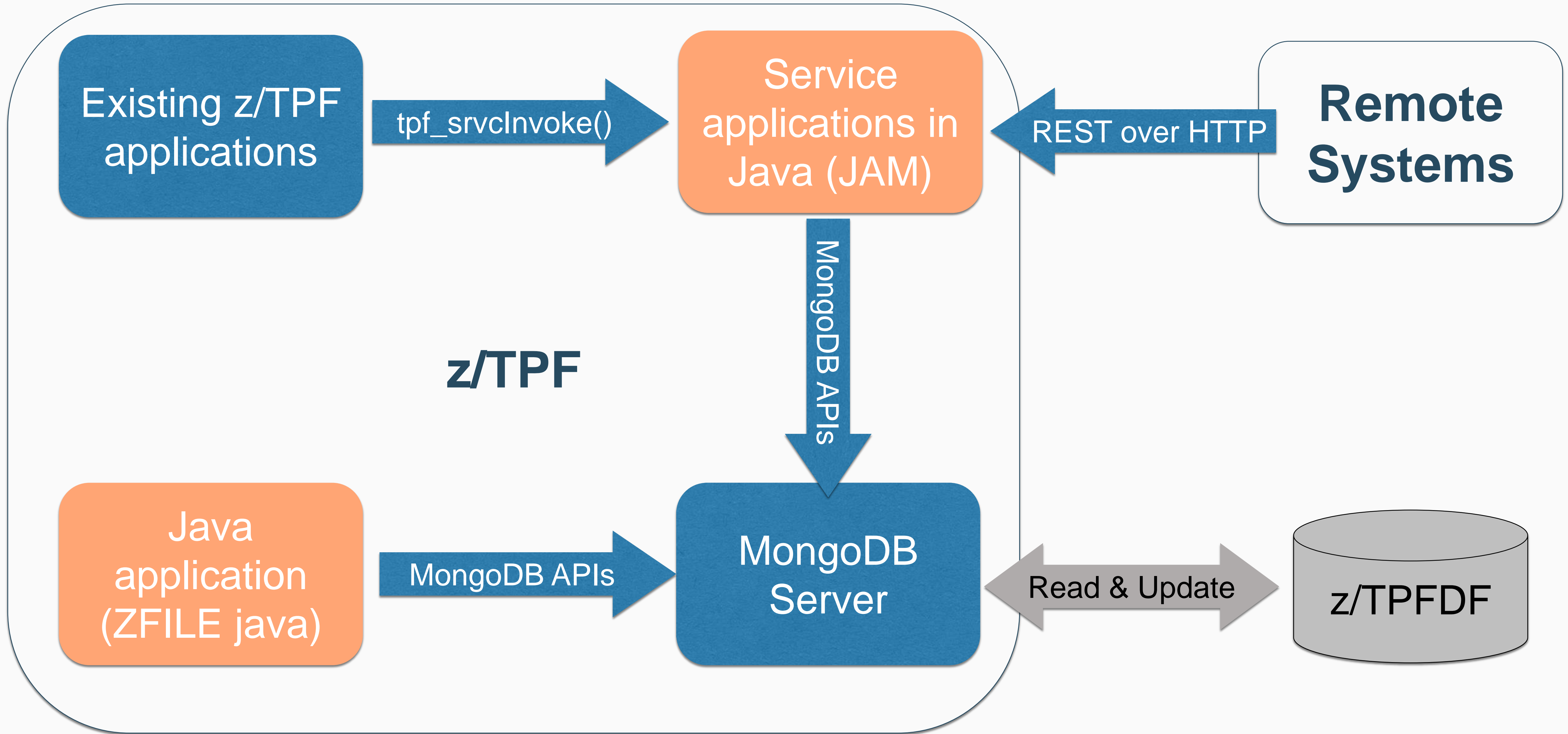
# Disclaimer

Any reference to future plans are for planning purposes only.

IBM reserves the right to change those plans at its discretion.

Any reliance on such a disclosure is solely at your own risk.

IBM makes no commitment to provide additional information in the future.





That's great, but...

- What if I need my Java program call existing z/TPF applications?
- How can I access my z/TPF FIND/FILE data from Java?
- What if I need to perform complex database updates or use commit scopes?





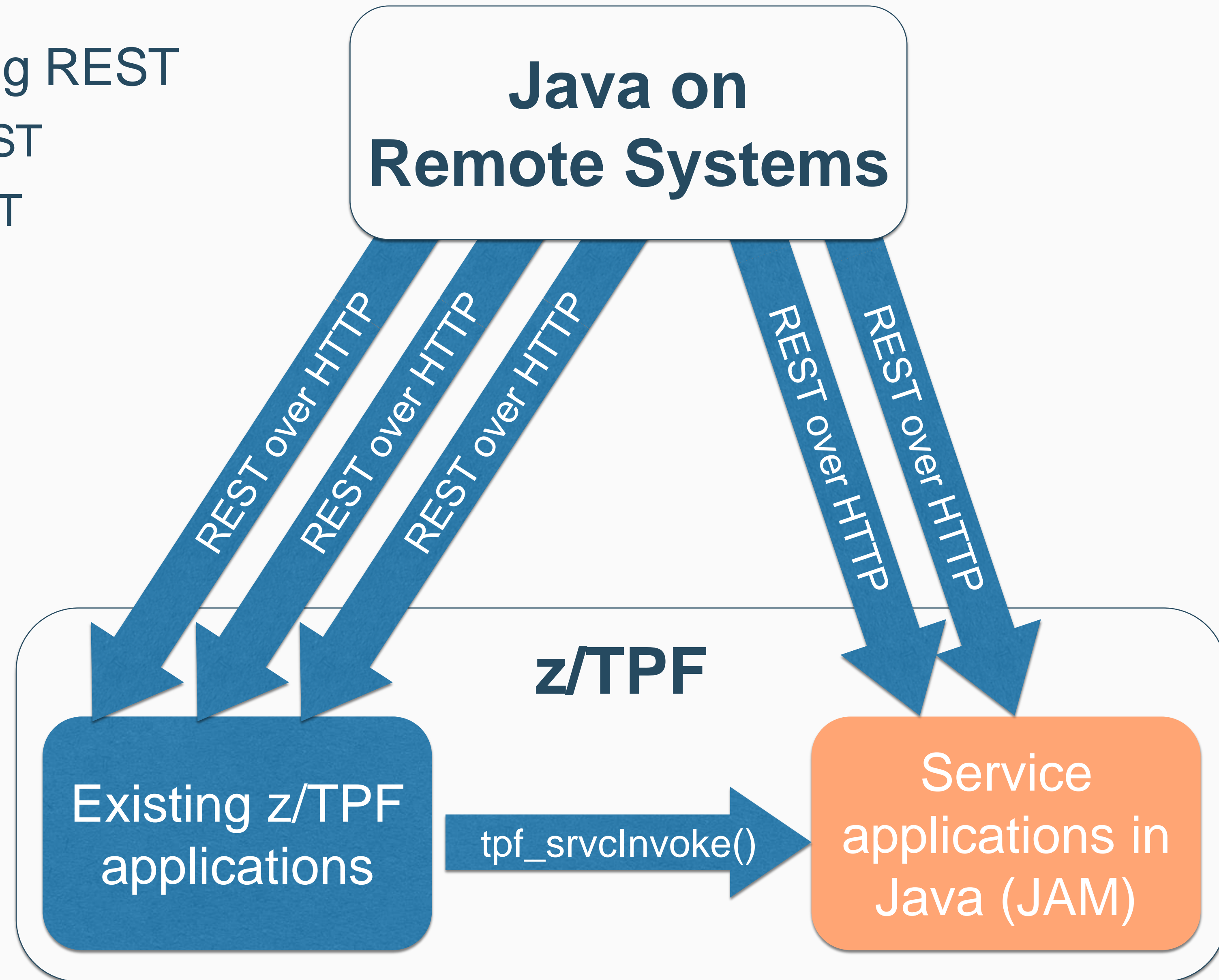
## Could I just deploy Java on a remote system?

Yes, you can call TPF services using REST

- Call existing TPF applications over REST
- Read and update databases over REST

But there are challenges...

- Network latency and overhead of communications stacks
- Scalability across remote systems and networks
- Complex database updates can be difficult
  - Make atomic database updates across REST?
  - Updates across multiple databases?
  - Commit scopes or compensating logic across REST?



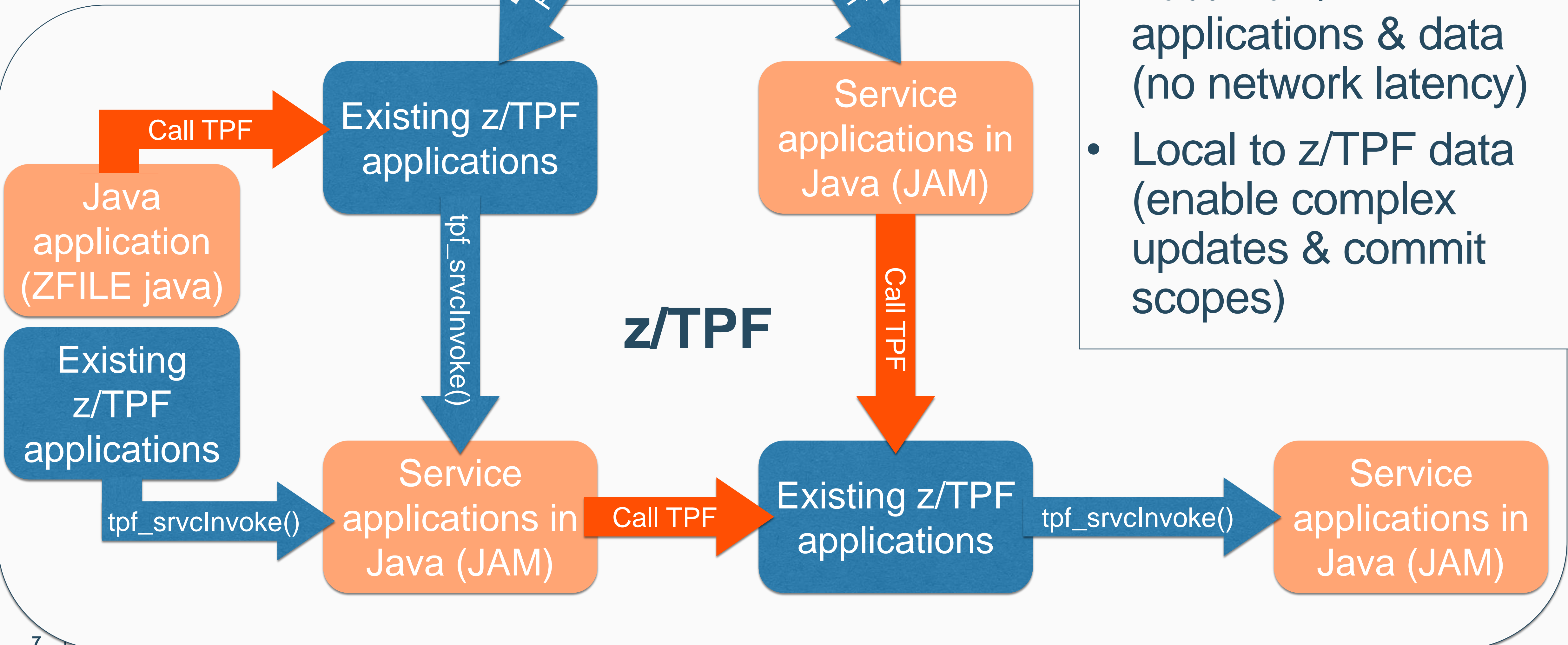
An **application architect** can use Java to incrementally modernize applications in place on z/TPF and **continue taking advantage of z/TPF's scalability and response times.**



Remote Systems

Put Java anywhere in the application stack!

- Local to z/TPF applications & data (no network latency)
- Local to z/TPF data (enable complex updates & commit scopes)





# But Java can call z/TPF today!

- Java Native Interface (JNI)
  - Usually means a Java programmer ...
    - Is writing C/C++ code
    - Interfacing with z/TPF application code
    - Translating between EBCDIC / Unicode and C structures / Java objects
  - z/TPF applications assume cleanup on exit
    - Java likely running as long running processes (JAMs) – no cleanup!
- REST Interface
  - Better...but goes through the communications stacks





# Java Subhill One

A **Java application programmer** can build a Java application and reuse existing business logic by calling z/TPF application services from Java **with the same effort as calling a REST service and without changing existing applications.**





# Java calling z/TPF – Programmer's View

Java  
Programmer

Call a REST  
Service

```
class pricingRequest
{
public:
    getFfid();
    setFfid();
    ...
    setAddons();

private:
    long ffid;
    String addons;

};
```

Java  
Process

z/TPF  
ECB

Convert  
Java  
Objects to  
C  
structures

```
struct pricingRequest
{
    unsigned int ffid;
    struct flightData
    {
        char departDate[6];
        char origCity[4];
        char departTime[6];
        char destCity[4];
    };
    char addons[3];
};
```

z/TPF  
Programmer

Create a  
REST service

Java application

```
PricingRequestApi Preq =
    JAXRSClientFactory.create(
        "http://localhost/pricingRequest",
        PricingRequestApi.class);

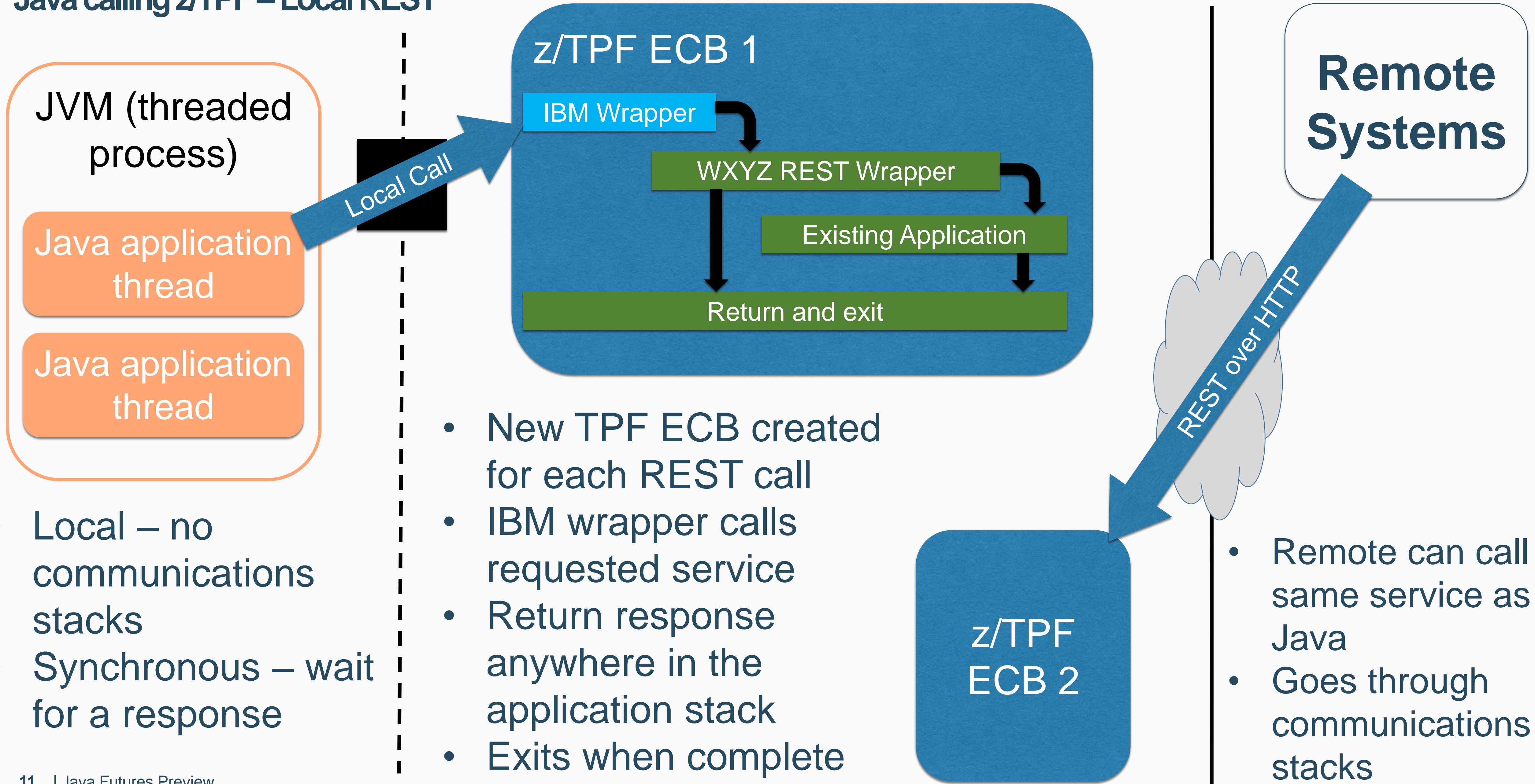
PricingResponse PRresp =
    Preq.pricingLookup(body);
```

z/TPF REST service wrapper WXYZ

```
void WXYZ (struct pricingRequest, length,
token)
{
    ...call existing z/TPF application
    logic
}
```



# Java calling z/TPF – Local REST



JVM (threaded process)

Java application thread

Java application thread

Local Call

z/TPF ECB 1

IBM Wrapper

WXYZ REST Wrapper

Existing Application

Return and exit

Remote Systems

REST over HTTP

z/TPF ECB 2

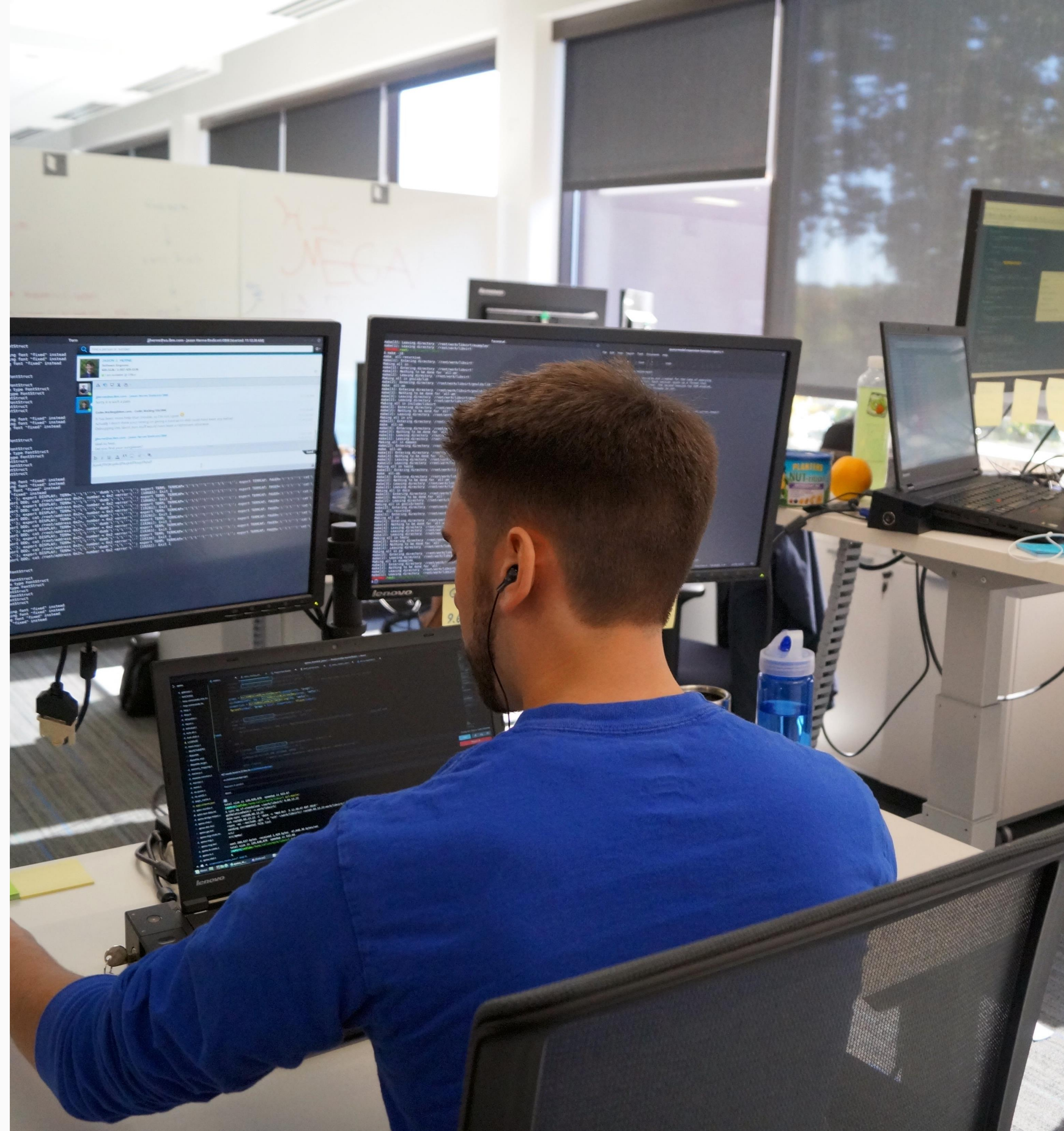
- New TPF ECB created for each REST call
- IBM wrapper calls requested service
- Return response anywhere in the application stack
- Exits when complete

- Remote can call same service as Java
- Goes through communications stacks



# Java Subhill Two

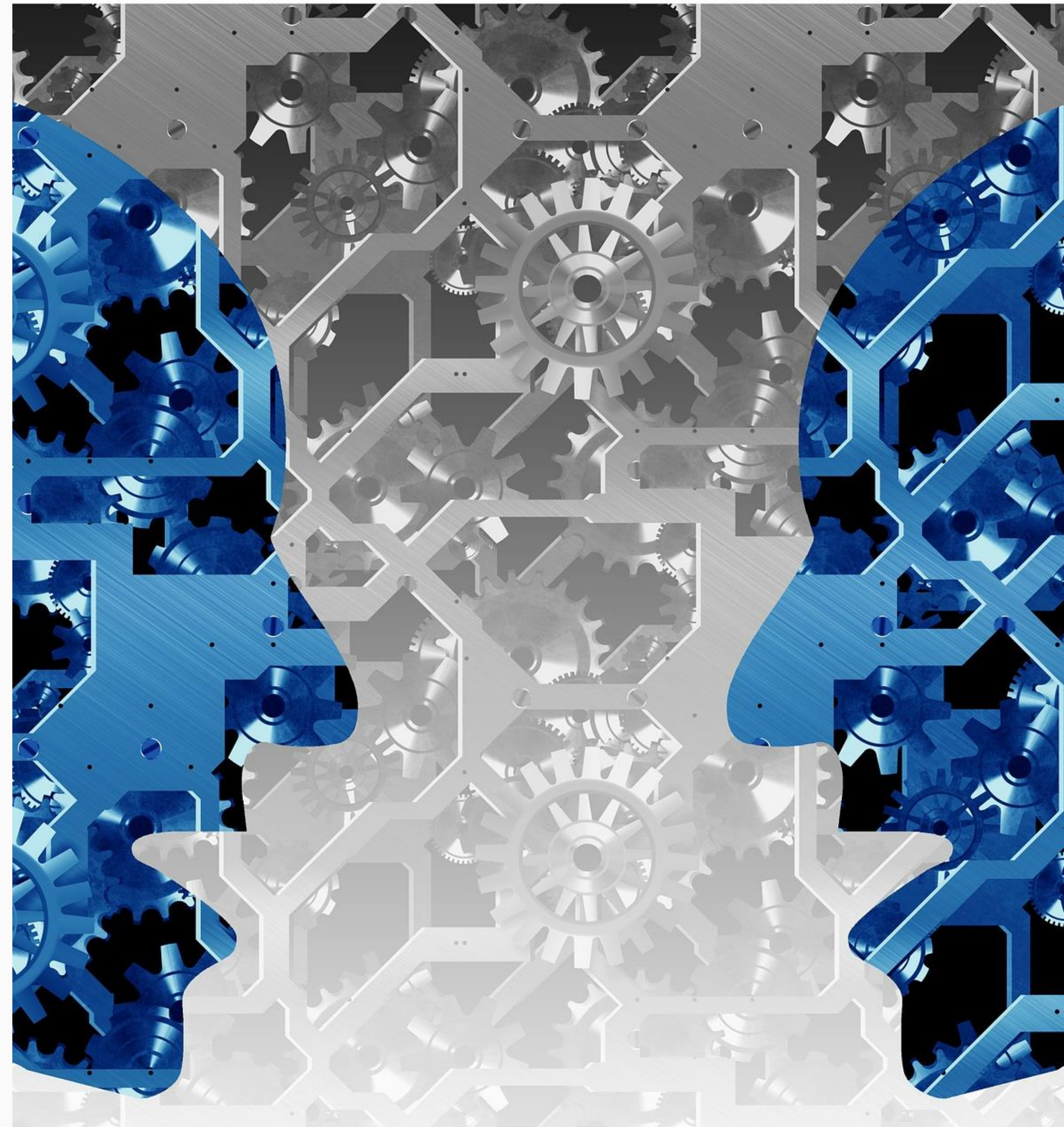
A **Java application programmer** can use a Java object model to access and update z/TPF databases in a consistent manner **without having to know z/TPF database constructs** and in half the time of legacy application programming models.





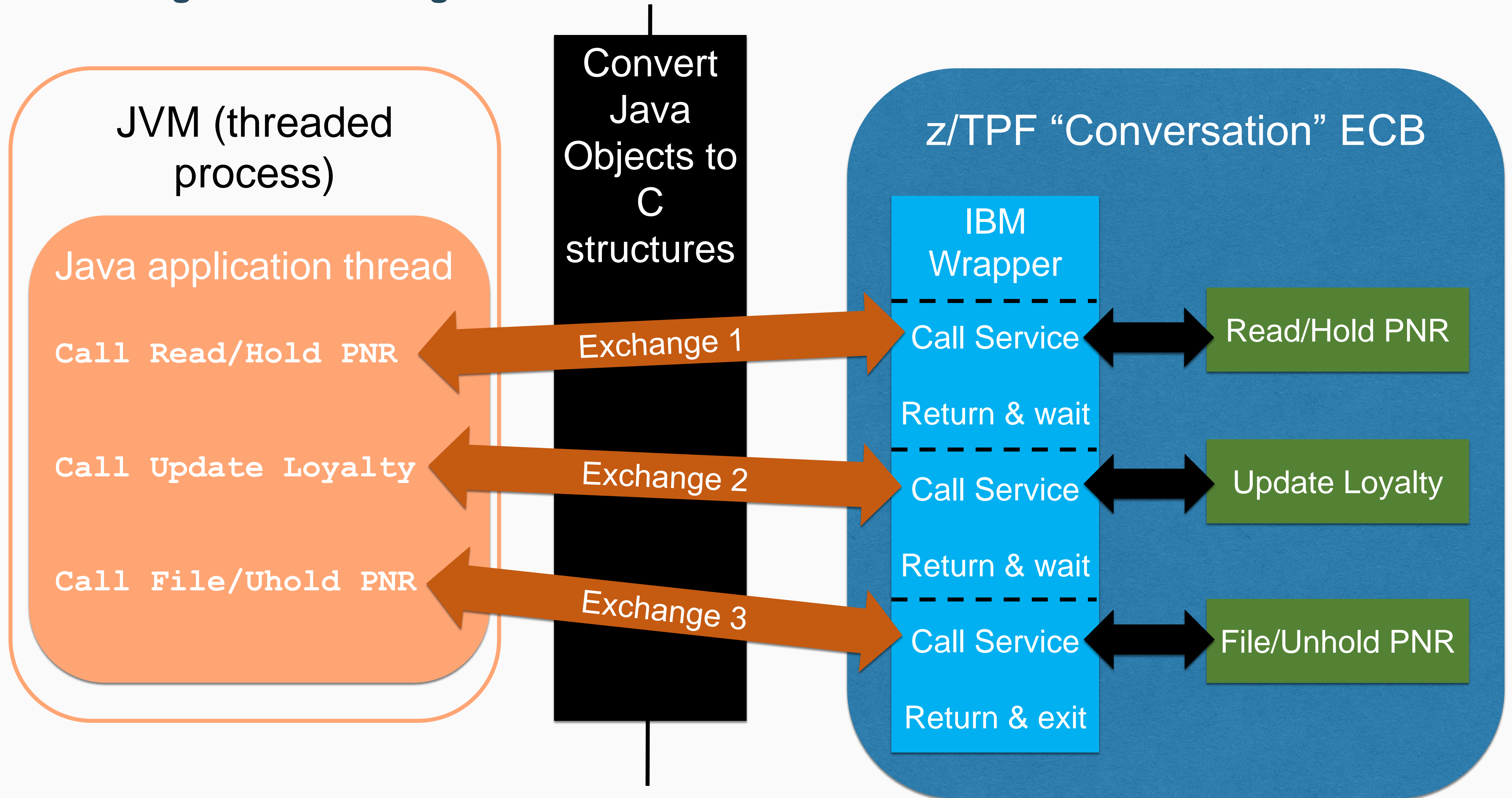
# Establishing a Conversation

- Define a conversation between a Java thread and a z/TPF ECB
  - z/TPF ECB created when Java starts the conversation
  - z/TPF ECB returns response but does not exit
  - Next exchange uses same z/TPF ECB
- z/TPF ECB can hold resources across exchanges and for the life of the conversation
  - Hold records
  - Open, commit, and rollback transaction scopes
  - Read and update multiple z/TPFDF and z/TPF databases





# Java accessing z/TPF data – Program flow





## How does coverage manage a Java environment?

- Existing TPF commands – ZDECB, ZSTAT, ZTRAP, ZFILE ps, etc.
  - Information for z/TPF system and individual ECBs
  - Limited insight into JVM or Java application
- Java traces and dumps
  - More information, but which JVM to signal for dumps/traces?
- Time consuming guess work to determine which JVM has issue and why

```

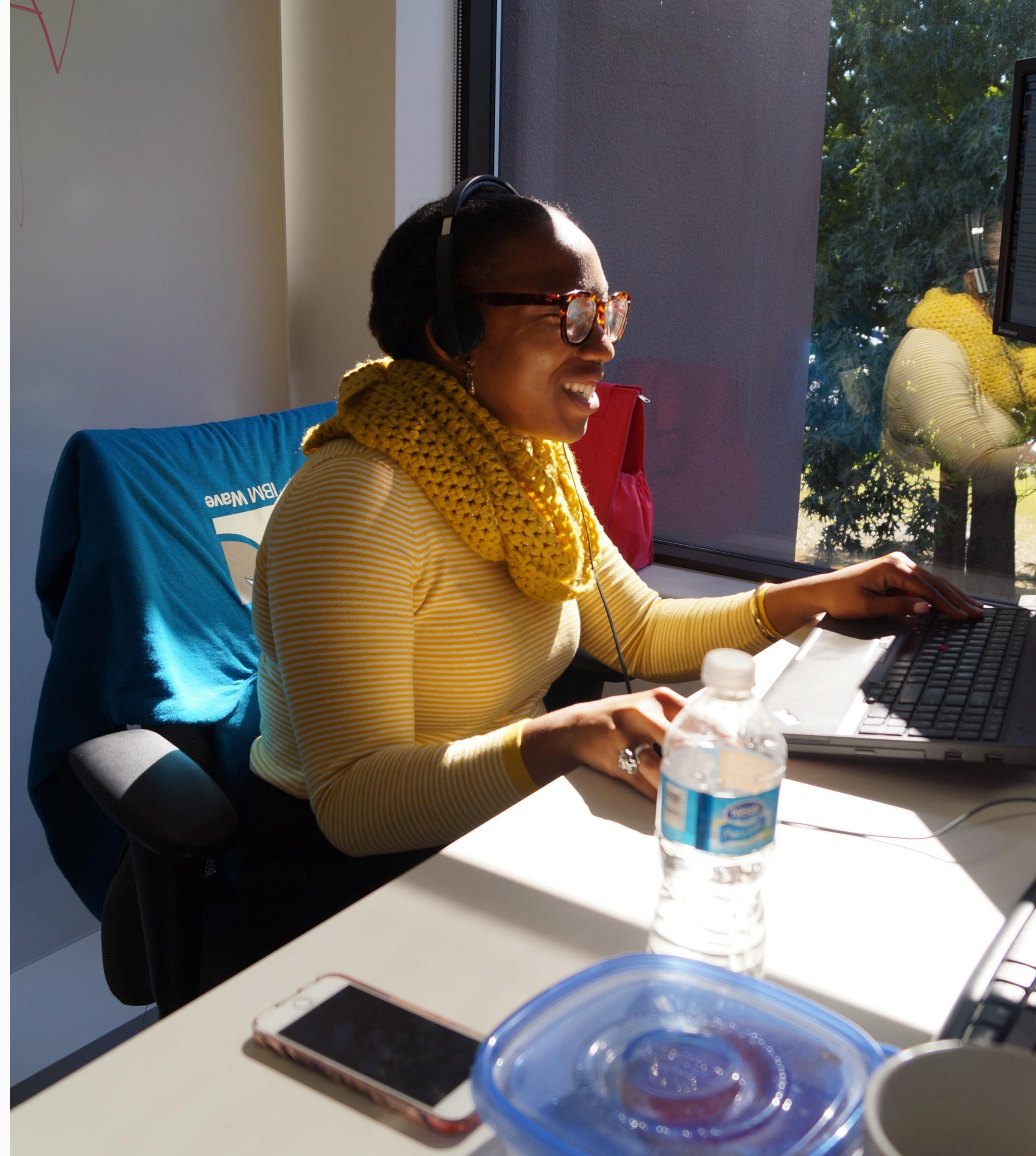
Edit View Format Save Areas Key Bindings Help
*PCOMM (Session A) DANBURY.CO0901 B_ X
CURRENT TOD MSTR IS NOT ACTIVE CPUID 'CPU 0FF325D37
SET THE TOD CLOCK USING FORMAT:
'ZATIM HHMM TOD ..... ' +
CLKS0010I 09.46.00 TIME OF DAY CLOCK LOCAL STANDARD TIME
TIME: 14.07.25 DATE: 03/28/17+
CLKS0005W 09.46.00 WAITING FOR CPC CLOCK CONFIRMATION+
CLKS0010I 09.46.00 TIME OF DAY CLOCK LOCAL STANDARD TIME
TIME: 14.07.33 DATE: 03/28/17+
CLKS0010I 09.46.00 TIME OF DAY CLOCK LOCAL STANDARD TIME
TIME: 14.07.33 DATE: 03/28/17+
TTCP0151E 09.46.33 REJECTED, OSA-OSA1 DOES NOT EXIST+
OSAE0020E 09.46.33 OSA-OSA1 DOES NOT EXIST+
CLKS0011A 09.46.33 BSS WILL CHANGE DATE WHEN CYCLED UP
ENTER ZATME GOOD TO CONTINUE CYCLE UP
OR
ENTER ZATME CNCL TO CANCEL CYCLE UP+

AAES0008I 00 ==> zdecb stats all
CSMP0097I 14.09.16 CPU-B SS-BSS SSU-HPN IS-01
DECB0014I 14.09.16 DISPLAY ECB SUMMARY
ECB ADDR SSU IS PGM TRC MIN SC MILS F4K F1MB FIND FILE GETF
10A42000 HPN 1 COMX COMX 59 37 18 1 188 11 0
10A1E000 HPN 1 CSL2 CSL2 59 12 3 0 90 0 0
10A2A000 HPN 1 COMX COMX 59 6 8 1 11 0 0
10A54000 HPN 1 COMX COMX 59 5 9 1 94 4 0
10A45000 HPN 1 CLTX CLTX 59 4 13 1 70 0 0
10A57000 HPN 1 CLTX CLTX 59 4 13 1 37 0 0
10A5D000 HPN 1 CLTX CLTX 59 4 13 1 72 0 0
10A87000 HPN 1 CSL2 CSL2 59 4 3 0 37 0 0
10A8A000 HPN 1 CJ00 CJ00 56 4 2 0 144 0 0
10A21000 HPN 1 CFVS CFVS 1 40 3 1 0 0 0
10A2D000 HPN 1 CLTX CLTX 59 3 13 1 36 0 0
10A06000 HPN 1 CLTX CLTX 59 2 4 1 0 0 0
10A0C000 HPN 1 COMX COMX 59 2 5 1 0 0 0
10A12000 HPN 1 CLTW CLTW 59 2 2 0 0 0 0
10A33000 HPN 1 CSL2 CSL2 59 2 16 2 4 0 0
10A39000 HPN 1 CLTZ CLTZ 59 2 7 1 13 0 0
MORE DATA AVAILABLE, ENTER ZPAGE TO CONTINUE+
AAES0008I 00 ==> zdsys
CSMP0097I 14.09.22 CPU-B SS-BSS SSU-HPN IS-01
DSYS0001I 14.09.22 THE SYSTEM IS IN NORM STATE FOR SUBSYSTEM BSS
ON CO0901 CPU-B 28MAR
END OF DISPLAY+
AAES0008I 00 ==> zfile java -version
CSMP0097I 14.09.35 CPU-B SS-BSS SSU-HPN IS-01
FILE0002I 14.09.35 START OF ERROR DISPLAY FROM java -version
java version "1.8.0"
Java(TM) SE Runtime Environment (build pxz6480sr3fp21-20170116_01(SR3 FP21))
IBM J9 VM (build 2.8, JRE 1.8.0 TPF s390x-64 20170118_333732 (JIT enabled, AOT
enabled)
J9VM - R28_Java8_SR3_20170118_1821_B333732
JIT - tr.r14.java.green_20161111_127476
GC - R28_Java8_SR3_20170118_1821_B333732
J9CL - 20161108_325074)
JCL - 20161111_01 based on Oracle jdk8u111-b14
END OF DISPLAY+
Ready. | INS | Server: 127.0.0.1 | HP: 1 | TA: 00 | SH: NA | FSC: 4000
    
```



# Java Subhill Three

Coverage programmers can monitor and be alerted to abnormal conditions in the Java environment and take corrective action within five minutes.





# IBM HealthCenter

- Monitoring and diagnostic tool
  - Agent runs on z/TPF and collects information on running JVMs
  - User interface runs as Eclipse plugin on Windows, Linux, Mac
- Use in development and production environments
  - Application activity (method profiling)
  - CPU and memory usage
  - Garbage collection statistics
- Currently available as [free download](#) for other platforms

The screenshot shows the IBM Support Assistant Workbench interface. The main window is titled "Method profile - IBM Support Assistant Workbench". It features a menu bar (File, Administration, Update, Data, Window, Help) and a toolbar with various icons. The interface is divided into several panes:

- Status:** Shows various system metrics with status indicators: Classes (info), Environment (check), Garbage Collection (warning), I/O (info), Locking (warning), Native Memory (check), Profiling (warning), and WebSphere Real Time (error).
- Method profile:** A table with columns: Samples, Self (%), Self, Tree (%), Tree, and Method. The top row is highlighted in red.
- Connection:** Shows connection details for "jsvtrt50:1972", including "378 MB received: Last updated 13:54:17" and a warning: "Some data was dropped because it was produced faster than the client could consume it. Around 8% of the data was lost."
- Analysis and Recommendations:** A warning icon and text: "The method LinkedList.get() is consuming approximately 25% of the CPU cycles. It may be a good candidate for optimization."
- Invocation paths:** A tree view showing "Methods that call LinkedList.get()":
  - java.util.LinkedList.get(int)
  - com.ibm.jtc.test.jltf.threads.RTThreadPlugin\$RTWorkThreadFactory.getNumberRunning() (100%)
    - com.ibm.jtc.test.jltf.threads.RTThreadPlugin\$RTWorkThreadFactory.getThreadStatus() (100%)
      - + com.ibm.jtc.test.jltf.ThreadPicker.getThreadStatus() (100%)



## Sponsor Users

- Sponsor user calls are in progress
- Early in design phase
- Contact us if you are interested in participating!







# THANK YOU

Questions or comments?

**Chris Filachek**

z/TPF and z/TPFDF Architecture & Development

© 2017 IBM z/TPF | TPF Users Group Spring Conference | IBM Confidential

IBM **z/TPF**  
April 4, 2017



IBM, the IBM logo, ibm.com and Rational are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at [“Copyright and trademark information”](http://www.ibm.com/legal/copytrade.shtml) at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

## Notes

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.